

[illegible]

OP
VO

.....

:

[illegible]

K 15
16-Sep-1984 02:28:09
14-Sep-1984 12:46:46

VAX-11 Bliss-32 V4.0-742
[MTAACP.SRC]OPRCOM.B32;1

Page 1
(1)

```
0001 0
0002 0 MODULE OPRCOM (LANGUAGE (BLISS32) ,
0003 0 IDENT = 'V04-000' ,
0004 0 ) =
0005 1 BEGIN
0006 1
0007 1 *****
0008 1 *
0009 1 * COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
0010 1 * DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
0011 1 * ALL RIGHTS RESERVED.
0012 1 *
0013 1 * THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
0014 1 * ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
0015 1 * INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
0016 1 * COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
0017 1 * OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
0018 1 * TRANSFERRED.
0019 1 *
0020 1 * THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
0021 1 * AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
0022 1 * CORPORATION.
0023 1 *
0024 1 * DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
0025 1 * SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
0026 1 *
0027 1 *
0028 1 *****
0029 1
0030 1 ++
0031 1
0032 1 FACILITY: MTAACP
0033 1
0034 1 ABSTRACT:
0035 1 This module communicates with the operator
0036 1
0037 1
0038 1 ENVIRONMENT:
0039 1
0040 1 VMS operating system, including privileged system services
0041 1 and internal exec routines.
0042 1
0043 1 --
0044 1
0045 1
0046 1
0047 1 AUTHOR: D. H. GILLESPIE, CREATION DATE: 24-AUG-1977
0048 1
0049 1 MODIFIED BY:
0050 1
0051 1 V03-002 HH0041 Hai Huang 24-Jul-1984
0052 1 Remove REQUIRE 'LIBD$:[VMSLIB.OBJ]MOUNTMSG.B32'.
0053 1
0054 1 V03-001 MMD0102 Meg Dumont, 17-Feb-1983 13:12
0055 1 Use GET_DEV_NAME to get the tape units' device name. Add
0056 1 check for OPCOM, so that the MTAACP can recover if OPCOM
0057 1 is not available.
```

```
.. 58      0058 1 |
.. 59      0059 1 |
.. 60      0060 1 |
.. 61      0061 1 |
.. 62      0062 1 |
.. 63      0063 1 |
.. 64      0064 1 |
.. 65      0065 1 |
.. 66      0066 1 |
.. 67      0067 1 |
.. 68      0068 1 |
.. 69      0069 1 |
.. 70      0070 1 |
.. 71      0071 1 |
.. 72      0072 1 |
.. 73      0456 1 |
.. 74      0457 1 |
.. 75      0458 1 |
.. 76      0459 1 |
.. 77      0460 1 |
.. 78      0461 1 |
.. 79      0462 1 |
.. 80      0463 1 |
.. 81      0464 1 |
.. 82      0465 1 |
.. 83      0466 1 |
.. 84      0467 1 |
.. 85      0468 1 |
.. 86      0469 1 |
.. 87      0470 1 |
.. 88      0471 1 |
.. 89      0472 1 |
.. 90      0473 1 |
.. 91      0474 1 |
.. 92      0475 1 |
.. 93      0476 1 |
.. 94      0477 1 |
.. 95      0478 1 |
.. 96      0479 1 |
.. 97      0480 1 |
.. 98      0481 1 |
.. 99      0482 1 |
.. 100     0483 1 |

V03-006 DMW00078      David Michael Walp      8-Feb-1982
Fixed pushing and popping of Username and Account CTL
fields around ooperator request.

V03-005 DMW00021      David Michael Walp      26-May-1981
Move GOLBAL LITERAL WORK_AREA_SZ to MTADEF.

V02-004 REFORMAT      Maria del C. Nasr      30-Jun-1980

**

LIBRARY 'SYS$LIBRARY:LIB.L32';

REQUIRE 'SRC$:MTADEF.B32';

FORWARD ROUTINE
CANCEL_OP_REPLY      : COMMON_CALL NOVALUE,      ! cancel reply from operator
FILL_PROC      : COMMON_CALL NOVALUE,
PRINT_NOT_LABEL      : NOVALUE JSB,      ! print not correct label
PRINT_OPR_MSG      : L$PRINT_OPR_MSG,      ! print operator message
SEND_MSG      : COMMON_CALL;      ! send msg to tape operator

GLOBAL
WORK_AREA      : VECTOR [WORK_AREA_SZ, BYTE];

EXTERNAL
CTL$T_USERNAME      : ADDRESSING_MODE (ABSOLUTE),
CTL$T_ACCOUNT      : ADDRESSING_MODE (ABSOLUTE),
CURRENT_UCB : REF BBLOCK,      ! address of current unit control block
MAIL_CHANNEL;      ! channel for reply mailbox

EXTERNAL ROUTINE
GET_DEV_NAME      : COMMON_CALL NOVALUE;      ! Given the UCB add get dev name

! SETUP MESSAGE DESCRIPTOR AND BUFFER IN WORK AREA

BIND
DESCR = WORK_AREA      : VECTOR [2],
MSG = WORK_AREA + 8 : BBLOCK;
```



```
102 0484 1 ROUTINE SEND_MSG (MB_CHANNEL) : COMMON_CALL =
103 0485 1
104 0486 1 ++
105 0487 1
106 0488 1 FUNCTIONAL DESCRIPTION:
107 0489 1 This routine formats and sends a message to the tape operator
108 0490 1
109 0491 1 CALLING SEQUENCE:
110 0492 1 SEND_MSG()
111 0493 1
112 0494 1 INPUT PARAMETERS:
113 0495 1 ARG1 - mailbox channel number
114 0496 1
115 0497 1 IMPLICIT INPUTS:
116 0498 1 CURRENT_VCB - address of current volume control block
117 0499 1 DESC[0] - length of text to be sent
118 0500 1 MSG[OPC$MS_TEXT] - text to be sent
119 0501 1
120 0502 1 OUTPUT PARAMETERS:
121 0503 1 NONE
122 0504 1
123 0505 1 IMPLICIT OUTPUTS:
124 0506 1 message sent
125 0507 1
126 0508 1 ROUTINE VALUE:
127 0509 1 $$$_DEVOFFLIN - no operator
128 0510 1 $$$_MBTOOSML - mailbox too small for message
129 0511 1
130 0512 1 SIDE EFFECTS:
131 0513 1 NONE
132 0514 1
133 0515 1 --
134 0516 1
135 0517 2 BEGIN
136 0518 2
137 0519 2 EXTERNAL REGISTER
138 0520 2 COMMON_REG;
139 0521 2
140 0522 2 EXTERNAL
141 0523 2 ACP_USERNAME : VECTOR [ , BYTE ],
142 0524 2 ACP_ACCOUNT : VECTOR [ , BYTE ];
143 0525 2
144 0526 2 BIND
145 0527 2 SECONDS = UPLIT (-10000000, -1);
146 0528 2
147 0529 2 LOCAL
148 0530 2 VVP : REF BBLOCK,
149 0531 2 STATUS ; ! status of mailbox write
150 0532 2
151 0533 2 ! format message header
152 0534 2
153 0535 2 MSG[OPC$MS_TARGET] = OPR$M_TAPES;
154 0536 2 MSG[OPC$MS_RPLYID] = .CURRENT_VCB; ! id if reply
155 0537 2 DESC[0] = .DESC[0] + $BYTEOFFSET(OPC$MS_TEXT);
156 0538 2 DESC[1] = MSG;
157 0539 2 VVP = .CURRENT_VCB [ VCB$MS_VPFL ];
158 0540 2
```

```

: 159 0541 2 INCR J FROM 0 TO 119 DO ! wait two minutes
: 160 0542 BEGIN
: 161 0543 ! make it look like the user is making the opcom request
: 162 0544 !
: 163 0545 !
: 164 0546 KERNEL_CALL ( FILL_PROC, VVP[VVP$T_USERNAME], VVP[VVP$T_ACCOUNT] );
: 165 0547
: 166 0548 STATUS = $SENDOPR ( MSGBUF = DESCR, CHAN = .MB_CHANNEL );
: 167 0549
: 168 0550 KERNEL_CALL ( FILL_PROC, ACP_USERNAME, ACP_ACCOUNT );
: 169 0551
: 170 0552 ! If the status is opc$_noperator then reset the error to be
: 171 0553 ! the fatal error. This means that there is no opcom available
: 172 0554 ! as would be the case if we are running on Midnite.
: 173 0555
: 174 0556 IF .STATUS EQL OPC$_NOPERATOR THEN
: 175 0557 BEGIN
: 176 0558 BIND
: 177 0559 STATUS2 = STATUS : BLOCK [4, BYTE];
: 178 0560 STATUS2 [ST$V_SEVERITY] = ST$K_ERROR;
: 179 0561 EXITLOOP;
: 180 0562
: 181 0563 END;
: 182 0564 ! Return to the caller if the send to operator worked. Or if it
: 183 0565 ! got an unrecoverable error. Device off line means there is
: 184 0566 ! no opcom available as would be the case if we are running on
: 185 0567 ! Midnite.
: 186 0568
: 187 0569 IF .STATUS
: 188 0570 OR
: 189 0571 (.STATUS NEQ SSS_INSFMEM
: 190 0572 AND
: 191 0573 .STATUS NEQ SSS_MBFULL)
: 192 0574 OR (.STATUS EQL SSS_DEVOFFLINE)
: 193 0575 THEN
: 194 0576 EXITLOOP; ! if temporary problem, then repeat
: 195 0577
: 196 0578 IF $SETIMR(DAYTIM = SECONDS, EFN = TIMEFN)
: 197 0579 THEN
: 198 0580 $WAITFR(EFN = TIMEFN);
: 199 0581
: 200 0582 END;
: 201 0583 RETURN .STATUS;
: 202 0584
: 203 0585
: 204 0586 1 END; ! end of routine
```

```

.TITLE OPRCOM
.IDENT \V04-000\
.PSECT $CODE$,NOWRT,2
.FFFFFFFF FF676980 00000 P.AAA: .LONG -10000000, -1
.PSECT $LOCKEDD1$,NOEXE,2
```


00000 WORK_AREA::
.BLKB 128DESCR=
MSG=
SECONDS=WORK_AREA
WORK_AREA+8
P.AAA.EXTRN CTLST USERNAME, CTLST ACCOUNT
.EXTRN CURRENT_UCB, MAIL_CHANNEL
.EXTRN GET_DEV_NAME, ACP_USERNAME
.EXTRN ACP_ACCOUNT, SYSSCMKRNL
.EXTRN SYSSNDOPR, SYSSSETIMR
.EXTRN SYSSWAITFR

.PSECT \$CODE\$,NOWRT,2

00FC 00000 SEND_MSG:

	57	00000000G	9F	9E	00002	.WORD	Save R2,R3,R4,R5,R6,R7	: 0484
	56	0000'	CF	9E	00009	MOVAB	@#SYSSCMKRNL, R7	:
09	A6		04	90	0000E	MOVAB	DESCR, R6	: 0535
0C	A6		5B	D0	00012	MOVB	#4, MSG+1	: 0536
	66		08	C0	00016	MOVL	CURRENT_VCB, MSG+4	: 0537
04	A6	08	A6	9E	00019	ADDL2	#8, DESCR	: 0538
	50	3C	AB	D0	0001E	MOVAB	MSG, DESCR+4	: 0539
	55	01BC	C0	9E	00022	MOVL	60(CURRENT_VCB), VVP	: 0546
	54	01B0	C0	9E	00027	MOVAB	444(R0), R5	:
			53	D4	0002C	MOVAB	432(R0), R4	:
			30	BB	0002E	CLRL	J	: 0548
			02	DD	00030	PUSHR	#^M<R4,R5>	: 0546
			5E	DD	00032	PUSHL	#2	:
		0000V	CF	9F	00034	PUSHL	SP	:
	67	04	05	FB	00038	PUSHAB	FILL PROC	:
			AC	DD	0003B	CALLS	#5, SYSSCMKRNL	:
			56	DD	0003E	PUSHL	MB_CHANNEL	: 0548
00000000G	00		02	FB	00040	PUSHL	R6	:
	52		50	D0	00047	CALLS	#2, SYSSNDOPR	:
		0000G	CF	9F	0004A	MOVL	R0, STATUS	:
		0000G	CF	9F	0004E	PUSHAB	ACP_ACCOUNT	: 0550
			02	DD	00052	PUSHAB	ACP_USERNAME	:
			5E	DD	00054	PUSHL	#2	:
		0000V	CF	9F	00056	PUSHL	SP	:
	67		05	FB	0005A	PUSHAB	FILL PROC	:
00058061	8F		52	D1	0005D	CALLS	#5, SYSSCMKRNL	:
			07	12	00064	CMPL	STATUS, #360545	: 0556
			02	F0	00066	BNEQ	2\$	:
52	03	00	41	11	0006B	INSV	#2, #0, #3, STATUS2	: 0560
			52	E8	0006D	BRB	5\$	: 0557
			52	D1	00070	BLBS	STATUS, 5\$	: 0569
00000124	8F		09	13	00077	CMPL	STATUS, #292	: 0571
			52	D1	00079	BEQL	3\$	:
000008D8	8F		2C	12	00080	CMPL	STATUS, #2264	: 0573
			52	D1	00082	BNEQ	5\$	:
00000084	8F		23	13	00089	CMPL	STATUS, #132	: 0574
			7E	7C	0008B	BEQL	5\$	:
		FF67	CF	9F	0008D	CLRQ	-(SP)	: 0578
			03	DD	00091	PUSHAB	SECONDS	:
00000000G	00		04	FB	00093	PUSHL	#3	:
						CALLS	#4, SYSSSETIMR	:

	09	50	E9 0009A	BLBC	RO, 4\$
		03	DD 0009D	PUSHL	#3
	00	01	FB 0009F	CALLS	#1, SYSS\$WAITFR
80 00000000G	53 00000077	8F	F3 000A6 4\$:	AOBLEQ	#119, J, 1\$
	50	52	D0 000AE 5\$:	MOVL	STATUS, RO
		04	000B1	RET	

0580
0541
0584
0586

; Routine Size: 178 bytes, Routine Base: \$CODE\$ + 0008

; 205 0587 1


```
207 0588 1 GLOBAL ROUTINE PRINT_NOT_LABEL (MVL_ENTRY) : JSB NOVALUE =
208 0589 1
209 0590 1 ++
210 0591 1
211 0592 1 FUNCTIONAL DESCRIPTION:
212 0593 1 This routine prints at the tape operator's terminal the message
213 0594 1 'vol on mtax: not xxxx'
214 0595 1
215 0596 1 CALLING SEQUENCE:
216 0597 1 PRINT_NOT_LABEL(ARG1)
217 0598 1
218 0599 1 INPUT PARAMETERS:
219 0600 1 NONE
220 0601 1
221 0602 1 IMPLICIT INPUTS:
222 0603 1 CURRENT_UCB - address of current unit control block
223 0604 1 CURRENT_VCB - address of current volume control block
224 0605 1
225 0606 1 OUTPUT PARAMETERS:
226 0607 1 NONE
227 0608 1
228 0609 1 IMPLICIT OUTPUTS:
229 0610 1 Message printed at tape operator's terminal unless
230 0611 1 there is no operator or the mailbox is too small
231 0612 1
232 0613 1 ROUTINE VALUE:
233 0614 1 NONE
234 0615 1
235 0616 1 SIDE EFFECTS:
236 0617 1 NONE
237 0618 1
238 0619 1 --
239 0620 1
240 0621 2 BEGIN
241 0622 2
242 0623 2 EXTERNAL REGISTER
243 0624 2 COMMON_REG;
244 0625 2
245 0626 2 LOCAL
246 0627 2 CVT_DEVNAM : VECTOR [MAX_DEVNAM_LENGTH, BYTE], ! Converted dev name
247 0628 2 CVT_DEVNAM_LENGTH : BYTE, ! and length of dev name
248 0629 2 LABEL_ADDR : REF VECTOR [, BYTE], ! address of label
249 0630 2 LABEL_SZ;
250 0631 2
251 0632 2 MAP
252 0633 2 MVL_ENTRY : REF BBLOCK; ! address of volume entry in mvl
253 0634 2
254 0635 2 LABEL_ADDR = MVL_ENTRY[MVL$T_VOLLBL];
255 0636 2 LABEL_SZ = 0;
256 0637 2
257 0638 2 ! flush out trailing blanks
258 0639 2 !
259 0640 2
260 0641 2 INCR I FROM 0 TO MVL$S_VOLLBL - 1 DO
261 0642 2 BEGIN
262 0643 2
263 0644 2 IF .LABEL_ADDR[I] EQL ' '
264 0644 2
```



```

: 264      0645      3      THEN
: 265      0646      3      EXITLOOP;
: 266      0647      3
: 267      0648      3      LABEL_SZ = .LABEL_SZ + 1;
: 268      0649      3      END;
: 269      0650      3
: 270      0651      3      ! This next call will use the UCB address to get the device's name and
: 271      0652      3      ! will fill in the fields with that name and the length of the name.
: 272      0653      3
: 273      0654      3      GET DEV_NAME(CVT_DEVNAM_LENGTH,CVT_DEVNAM);
: 274      0655      3      PRINT_OPR_MSG(MOONS$ NOT LABEL, 0, .LABEL_SZ, .LABEL_ADDR,
: 275      0656      3      .CVT_DEVNAM_LENGTH,CVT_DEVNAM);
: 276      0657      1      END;                                ! end of routine
```

			0C	BB	00000	PRINT_NOT_LABEL::		
						PUSHR	#^M<R2,R3>	: 0588
						SUBL2	#20, SP	
						MOVL	MVL_ENTRY, LABEL_ADDR	: 0635
						CLRL	LABEL_SZ	: 0636
						CLRL	I	: 0644
						CMPB	(I)[LABEL_ADDR], #32	
						BEQL	2\$	
						INCL	LABEL_SZ	: 0648
						AOBLEQ	#5, I, 1\$	: 0641
						PUSHAB	CVT_DEVNAM	: 0654
						PUSHAB	CVT_DEVNAM_LENGTH	
						CALLS	#2, GET_DEV_NAME	
						PUSHAB	CVT_DEVNAM	: 0655
						MOVZBL	CVT_DEVNAM_LENGTH, -(SP)	: 0656
						PUSHL	LABEL_ADDR	: 0655
						PUSHL	LABEL_SZ	
						CLRL	-(SP)	
						PUSHL	#7504132	
						BSBW	PRINT_OPR_MSG	
						ADDL2	#44, SP	: 0657
						POPR	#^M<R2,R3>	
						RSB		

5E 14 C2 00002
52 20 AE D0 00005
53 D4 00009
50 D4 0000B
20 6042 91 0000D 1\$:
06 13 00011
53 D6 00013
50 05 F3 00015
04 AE 9F 00019 2\$:
04 AE 9F 0001C
0000G CF 02 FB 0001F
04 AE 9F 00024
7E 04 AE 9A 00027
52 DD 0002B
53 DD 0002D
7E D4 0002F
00728104 8F DD 00031
5E 0000V 30 00037
2C C0 0003A
0C BA 0003D
05 0003F

; Routine Size: 64 bytes, Routine Base: \$CODE\$ + 00BA

; 277 0658 1


```
: 279      0659 1 GLOBAL ROUTINE CANCEL_OP_REPLY : COMMON_CALL NOVALUE =
: 280      0660 1
: 281      0661 1 ++
: 282      0662 1
: 283      0663 1 FUNCTIONAL DESCRIPTION:
: 284      0664 1     This routine cancels a request sent to the operator
: 285      0665 1
: 286      0666 1 CALLING SEQUENCE:
: 287      0667 1     CANCEL_OP_REPLY()
: 288      0668 1
: 289      0669 1 INPUT PARAMETERS:
: 290      0670 1     NONE
: 291      0671 1
: 292      0672 1 IMPLICIT INPUTS:
: 293      0673 1     CURRENT_VCB - address of current volume control block
: 294      0674 1
: 295      0675 1 OUTPUT PARAMETERS:
: 296      0676 1     NONE
: 297      0677 1
: 298      0678 1 IMPLICIT OUTPUTS:
: 299      0679 1     NONE
: 300      0680 1
: 301      0681 1 ROUTINE VALUE:
: 302      0682 1     NONE
: 303      0683 1
: 304      0684 1 SIDE EFFECTS:
: 305      0685 1     NONE
: 306      0686 1
: 307      0687 1 USER ERRORS:
: 308      0688 1     NONE
: 309      0689 1
: 310      0690 1 --
: 311      0691 1
: 312      0692 2 BEGIN
: 313      0693 2
: 314      0694 2 EXTERNAL REGISTER
: 315      0695 2     COMMON_REG;
: 316      0696 2
: 317      0697 2 MSG[OPCSB_MS_TYPE] = OPCS_RQ_CANCEL;
: 318      0698 2 DESCR[0] = 0;
: 319      0699 2 RETURN SEND_MSG(.MAIL_CHANNEL);
: 320      0700 2
: 321      0701 1 END;
```

! no message text

```
0000' CF          0000 0000
                   05 90 00002
0000' CF D4 00007
0000G CF DD 0000B
FEFA CF          01 FB 000CF
                   04 00014
```

```
.ENTRY CANCEL_OP_REPLY, Save nothing
MOVB #5, MSG
CLRL DESCR
PUSHL MAIL_CHANNEL
CALLS #1, SEND_MSG
RET
```

```
: 0659
: 0697
: 0698
: 0699
: 0701
```

; Routine Size: 21 bytes, Routine Base: \$CODES + 00FA

OPRCOM
V04-000

G 16
16-Sep-1984 02:28:09
14-Sep-1984 12:46:46

VAX-11 Bliss-32 V4.0-742
[MTAACP.SRC]OPRCOM.B32;1

Page 10
(4)

; 322

0702 1


```

324 0703 1 GLOBAL ROUTINE PRINT_OPR_MSG (MSGNO, CHANNEL, ARGLIST) : L$PRINT_OPR_MSG =
325 0704 1
326 0705 1 !++
327 0706 1
328 0707 1 FUNCTIONAL DESCRIPTION:
329 0708 1 This routine gets a system message, formats the messages and
330 0709 1 prints it.
331 0710 1
332 0711 1 CALLING SEQUENCE:
333 0712 1 PRINT_OPR_MSG(ARG1,ARG2,ARG3)
334 0713 1
335 0714 1 INPUT PARAMETERS:
336 0715 1 ARG1 - system message number
337 0716 1 ARG2 - channel number for reply if one is wanted
338 0717 1 ARG3 - arguments for inclusion in message
339 0718 1
340 0719 1 IMPLICIT INPUTS:
341 0720 1 NONE
342 0721 1
343 0722 1 OUTPUT PARAMETERS:
344 0723 1 NONE
345 0724 1
346 0725 1 IMPLICIT OUTPUTS:
347 0726 1 NONE
348 0727 1
349 0728 1 ROUTINE VALUE:
350 0729 1 status code from send to operator
351 0730 1
352 0731 1 SIDE EFFECTS:
353 0732 1 DATA is used as a work area(also used by RATTR)
354 0733 1
355 0734 1 --
356 0735 1
357 0736 2 BEGIN
358 0737 2
359 0738 2 EXTERNAL REGISTER
360 0739 2 COMMON_REG;
361 0740 2
362 0741 2 EXTERNAL ROUTINE
363 0742 2 SY$FAOL : ADDRESSING_MODE (ABSOLUTE),
364 0743 2 SY$GETMSG : ADDRESSING_MODE (ABSOLUTE);
365 0744 2
366 0745 2 EXTERNAL
367 0746 2 DATA;
368 0747 2
369 0748 2 LOCAL
370 0749 2 DESCR_FAOSTR : VECTOR [2];
371 0750 2
372 0751 2 DESCR_FAOSTR[0] = 256;
373 0752 2 DESCR_FAOSTR[1] = DATA;
374 0753 2 SY$GETMSG(MSGNO, DESCR_FAOSTR, DESCR_FAOSTR, 1, 0);
375 0754 2 DESCR[0] = WORK AREA SZ = 8 - $BYTEOFFSET(OPC$L_MS_TEXT);
376 0755 2 DESCR[1] = MSG[OPC$L_MS_TEXT];
377 0756 2 SY$FAOL(DESCR_FAOSTR, DESCR, DESCR, ARGLIST);
378 0757 2 MSG[OPC$B_MS_TYPE] = OPC$_RQ_RQST;
379 0758 2 (
380 0759 3
```

```
.. 381      0760 3
: 382      0761
: 383      0762
: 384      0763
: 385      0764
: 386      0765
: 387      0766
: 388      0767
: 389      0768
: 390      0769
: 391      0770
: 392      0771 1
```

```
LOCAL
  STATUS;

STATUS = SEND_MSG(.CHANNEL);

IF NOT .STATUS
THEN
  RETURN .STATUS);

RETURN 1;

END;
```

! end of routine PRINT_OPR_MSG

```
                                .EXTRN SYSS$FAOL, SYSS$GETMSG
                                .EXTRN DATA

                                04 C2 00000 PRINT_OPR MSG::
                                04 7E 0100 8F 3C 00003
                                04 AE 0000G CF 9E 00008
                                7E 01 7D 0000E
                                08 AE 9F 00011
                                0C AE 9F 00014
                                1C AE DD 00017
                                00000000G 9F 05 FB 0001A
                                0000' CF 70 8F 9A 00021
                                0000' CF 0000' CF 9E 00027
                                14 AE 9F 0002E
                                0000' CF 9F 00031
                                0000' CF 9F 00035
                                0C AE 9F 00039
                                00000000G 9F 04 FB 0003C
                                0000' CF 03 90 00043
                                10 AE DD 00048
                                FEA9 CF 01 FB 0004B
                                03 50 E9 00050
                                50 01 D0 00053
                                5E 08 C0 00056 1$:
                                05 00059
                                04 7E 0100 8F 3C 00003
                                04 AE 0000G CF 9E 00008
                                7E 01 7D 0000E
                                08 AE 9F 00011
                                0C AE 9F 00014
                                1C AE DD 00017
                                00000000G 9F 05 FB 0001A
                                0000' CF 70 8F 9A 00021
                                0000' CF 0000' CF 9E 00027
                                14 AE 9F 0002E
                                0000' CF 9F 00031
                                0000' CF 9F 00035
                                0C AE 9F 00039
                                00000000G 9F 04 FB 0003C
                                0000' CF 03 90 00043
                                10 AE DD 00048
                                FEA9 CF 01 FB 0004B
                                03 50 E9 00050
                                50 01 D0 00053
                                5E 08 C0 00056 1$:
                                05 00059

                                SUBL2 #4, SP
                                MOVZWL #256, DESCR_FAOSTR
                                MOVAB DATA, DESCR_FAOSTR+4
                                MOVQ #1, -(SP)
                                PUSHAB DESCR_FAOSTR
                                PUSHAB DESCR_FAOSTR
                                PUSHL MSGNO
                                CALLS #5, @SYSS$GETMSG
                                MOVZBL #112, DESCR
                                MOVAB MSG+8, DESCR+4
                                PUSHAB ARGLIST
                                PUSHAB DESCR
                                PUSHAB DESCR
                                PUSHAB DESCR_FAOSTR
                                CALLS #4, @SYSS$FAOL
                                MOVAB #3, MSG
                                PUSHL CHANNEL
                                CALLS #1, SEND_MSG
                                BLBC STATUS, T$
                                MOVL #1, R0
                                ADDL2 #8, SP
                                RSB
```

```
.. 0703
: 0751
: 0752
: 0753
:
:
:
: 0754
: 0755
: 0756
:
:
: 0757
: 0763
:
: 0765
: 0769
: 0771
:
```

; Routine Size: 90 bytes, Routine Base: \$CODE\$ + 010F

; 393 0772 1


```

: 395      0773 1 ROUTINE FILL_PROC ( USERNAME, ACCOUNT ) : COMMON_CALL NOVALUE =
: 396      0774 1
: 397      0775 1 |++
: 398      0776 1
: 399      0777 1 FUNCTIONAL DESCRIPTION:
: 400      0778 1     This routine fills in the username and account in P1 space. So
: 401      0779 1     Opcom messages look like they are coming from the user rather than
: 402      0780 1     ACP
: 403      0781 1
: 404      0782 1 CALLING SEQUENCE:
: 405      0783 1     FILE_PROC ( AGR1, ARG2 ) in kernel mode
: 406      0784 1
: 407      0785 1 INPUT PARAMETERS:
: 408      0786 1     ARG1 - address of username string
: 409      0787 1     ARG2 - address of account name string
: 410      0788 1
: 411      0789 1 IMPLICIT INPUTS:
: 412      0790 1     NONE
: 413      0791 1
: 414      0792 1 OUTPUT PARAMETERS:
: 415      0793 1     NONE
: 416      0794 1
: 417      0795 1 IMPLICIT OUTPUTS:
: 418      0796 1     NONE
: 419      0797 1
: 420      0798 1 ROUTINE VALUE:
: 421      0799 1     NONE
: 422      0800 1
: 423      0801 1 SIDE EFFECTS:
: 424      0802 1     NONE
: 425      0803 1
: 426      0804 1 |--
: 427      0805 1
: 428      0806 2 BEGIN
: 429      0807 2
: 430      0808 2 MAP
: 431      0809 2     USERNAME : REF VECTOR [ , BYTE ],
: 432      0810 2     ACCOUNT  : REF VECTOR [ , BYTE ];
: 433      0811 2
: 434      0812 2     CH$MOVE ( VVP$$_USERNAME, .USERNAME, CTL$_USERNAME);
: 435      0813 2     CH$MOVE ( VVP$$_ACCOUNT, .ACCOUNT, CTL$_ACCOUNT);
: 436      0814 1 END;
```

```

                                003C 00000 FILL_PROC:
                                .WORD      Save R2,R3,R4,R5
00000000G 9F      04 BC      0C 28 00002      MOV C3 #12, @USERNAME, @CTL$_USERNAME
00000000G 9F      08 BC      08 28 0000B      MOV C3 #8, @ACCOUNT, @CTL$_ACCOUNT
                                04 00014      RET
```

```

: 0773
: 0812
: 0813
: 0814
```

; Routine Size: 21 bytes, Routine Base: \$CODE\$ + 0169

; 437 0815 1 END

OPRCOM
V04-000

K 16
16-Sep-1984 02:28:09
14-Sep-1984 12:46:46

VAX-11 Bliss-32 V4.0-742
[MTAACP.SRC]OPRCOM.B32;1

Page 14
(6)

: 438 0816 1
: 439 0817 0 ELUDOM

PSECT SUMMARY

Name	Bytes	Attributes
\$LOCKEDD1\$	128 NOVEC, WRT, RD ,NOEXE,NOSHR,	LCL, REL, CON,NOPIC,ALIGN(2)
\$CODE\$	382 NOVEC,NOWRT, RD , EXE,NOSHR,	LCL, REL, CON,NOPIC,ALIGN(2)

Library Statistics

File	----- Total	Symbols Loaded	----- Percent	Pages Mapped	Processing Time
_\$255\$DUA28:[SYSLIB]LIB.L32;1	18619	26	0	1000	00:01.8

COMMAND QUALIFIERS

: BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/LIS=LIS\$:OPRCOM/OBJ=OBJ\$:OPRCOM MSRC\$:OPRCOM/UPDATE=(ENH\$:OPRCOM)

: 440 0818 0
: Size: 374 code + 136 data bytes
: Run Time: 00:11.0
: Elapsed Time: 00:23.4
: Lines/CPU Min: 4453
: Lexemes/CPU-Min: 19655
: Memory Used: 105 pages
: Compilation Complete

0255 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

